IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

| | | | |
|---|---|---|---|
| Applicant(s): | Ryan Shillington, et al. | | |
| Assignee: | Versata Development Group, Inc. | | |
| Title: | SYSTEM AND METHOD FOR REMOTELY DEBUGGING APPLICATION PROGRAMS | | |
| Serial No.: | 09/902,128 | Filed: | July 10, 2001 |
| Examiner: | Yolanda L. Wilson | Group Art Unit: | 2184 |
| Docket No.: | T00053 | Customer No.: | 33438 |

Austin, Texas
October 18, 2007

ELECTRONICALLY FILED

**APPEAL BRIEF UNDER 37 CFR § 41.37**

Dear Sir:

Applicant submits this Appeal Brief pursuant to the Notice of Appeal filed in this case on March 6, 2007, and the Notice of Panel Decision from Pre-Appeal Brief Review dated April 18, 2007 with an extendible Appeal Brief due date of May 18, 2007. The Appeal Brief fee is being paid via the USPTO EFS. The Board is authorized to deduct any other amounts required for this appeal brief and to credit any amounts overpaid to Deposit Account No. 502264. Accompanying this Appeal Brief is a Petition for Extension of time for five (5) months setting a new time for response of October 18, 2007.

Also concurrently filed with this Appeal Brief is a *Petition to Withdraw a Holding of Abandonment* pursuant to 37 CFR § 1.181(a).

**REAL PARTY IN INTEREST - 37 CFR § 41.37(c)(1)(i)**

The real party in interest is the assignee, Versata Development Group, Inc., as evidenced by the assignment set forth at Reel/Frame 019035/0545.

**RELATED APPEALS AND INTERFERENCES - 37 CFR § 41.37(c)(1)(ii)**

There are no appeals or interferences that could directly affect or be directly affected by or have a bearing on the decision by the Board of Patent Appeals and Interferences in the pending appeal.

**STATUS OF CLAIMS - 37 CFR § 41.37(c)(1)(iii)**

- Claims 1-44 are pending.
- Claims 1-44 stand finally rejected as set forth in the Final Office Action dated September 6, 2006.
- The rejection of claims 1-44 is appealed.
- Appendix "A" contains the full set of pending claims.

**STATUS OF AMENDMENTS - 37 CFR § 41.37(c)(1)(iv)**

No amendments after final have been requested or entered.

**SUMMARY OF CLAIMED SUBJECT MATTER - 37 CFR § 41.37(c)(1)(v)**

A concise explanation of the subject matter defined in each of the independent claims involved in the appeal and each dependent claim argued separately is set forth below. References to the specification recite appropriate page numbers and paragraph numbers as set forth in the specification filed on July 10, 2001. Applicants specifically note that the present invention is defined by the claims and not by specific embodiments set forth in the Detailed Description of the specification.

**Claim 1.**

With reference to the example embodiments of Figures 3, 4a, 4b, 4c, and 4d of the present application, the invention of claim 1 provides a method of debugging an application program residing on a server using a debugger program, also residing on the server, and a web browser executed at a workstation. The "server [] is remote from the workstation." Claim 1.

Referring to Figure 3, a web browser 314 is executed on a client workstation 112. An application program 304 executes on a server 302. Present Application, p. 7, lines 27-29. A debugger program 312 also executes on server 302. *Id.*, p. 8, lines 5-11. The client workstation 112 and server 302 are remote from each other and connected to each other via a network interface 122. *Id.*, p. 6, lines 8-9. "Web browser 314 provides a user interface to access application programs 304 and debugger program 312 via network interface 122." *Id.*, p. 8, lines 7-8. "After [a] user enters [a uniform resource locator] into the workstation 112, the web browser 314 contacts the server via network interface 122, the corresponding application

program 304 generates the requested web page, and the requested web page is transmitted via the network interface 122 to the web browser 314." *Id.*, lines 14-18. "Web browser 314 presents the requested web page to the user." *Id.*, line 18.

Referring to Figures 4a-d, the "GUI display 400 includes selectable features, such as tabs 404 and 406 to provide options to display a user frame 408 or a debugger frame 410." *Id.*, p. 9, lines 1-2. The "user frame 408 includes information generated by the application program 304 being debugged." *Id.*, lines 3-4. "For example, Fig. 4a shows a catalog web page for a color monitor in user frame 408." *Id.*, lines 6-7.

"For one of many reasons, user frame 408 may not render correctly." *Id.*, line 11. "To investigate the reasons for the problem, the developer selects the debugger view tab 406, thereby causing the web browser 314 to link to the debugger program 312." *Id.*, lines 11-13. "The debugger program 312 generates the debug information displayed in the debugger frame 410." *Id.*, lines 14-15. "Figs. 4b-d show the debugger frame 410 with the various components in the application program 304, such as variables, variable values, fields, method calls, and constructors, used to generate the catalog web page I user frame 408" when the debug view option is selected. *Id.*, lines 7-9. "A user can investigate the source of errors in the application program 304 that generates the information or error messages on the user frame 408." *Id.*, lines 15-16. "For example, user frame 408 can display the generic message "Null Pointer Exception" generated by the application program 304 when one or more variable(s) used by the web page has a null value." *Id.*, lines 16-19. "The debugger frame 410 allows the user to display the types and values of variables, methods and constructors associated with the variables, as shown in Figs 4b-4d." *Id.*, lines 19-21.

**Claim 9.**

With reference to the example embodiments of Figures 3, 4a, 4b, 4c, 4d, and 5 of the present application, the invention of claim 9 provides a "method of debugging an application program from a workstation, wherein the application program and a debugger program reside on a server that is remote from the workstation." Claim 9.

Referring to Figure 3, an application program 304 executes on a server 302. Present Application, p. 7, lines 27-29. "Server 302 also includes a debugger program 312 that can be used to troubleshoot problems from one or more remote client workstations 112." *Id.*, p. 8, lines

5-6. The debugger program 312 also executes on server 302. *Id.*, p. 8, lines 5-11. A "user frame includes information generated by the application program 304 being debugged." *Id.*, 9, lines 3-4.

A debugger program "allows access to information pertaining to the application program being debugged." Page 5, lines 17-18. An example of when debugger 312 can be used is when a user, running web browser 314 on the workstation 112, enters the universal resource locator (URL) to access one of the application programs 304 for a web page." *Id.*, p. 8, lines 12-14. "After the user enters the URL into the workstation 12, the web browser 314 contacts the server 302 via the network interface 122, the corresponding application program 304 generates the requested web page, and the requested web page is transmitted via the network interface 122 to the web browser 314." *Id.*, lines 14-18. "The web page can include information from internal code 305, private code 308, and/or public code 310." *Id.*, lines 18-20.

"The debugger program 312 generates the debug information displayed in the debugger frame 410" that is transmitted to the workstation. *Id.*, p. 9, lines 14-15. Referring to Figures 3, 4b-4d, and 5, "Fig. 5 shows an embodiment of a process for generating the debugger frame 410 (Figs. 4b-4d) for an application program 304 (Fig. 3)." *Id.*, p. 9, lines 26-27. "The debugger frame 410 is mapped to the user frame 408 from the standpoint that the context of the debugger frame 410 is sensitive to the context of the user frame 408." *Id.*, lines 4-6. "In process 502, the current state of the application program 304 is stored." *Id.*, line 28. "Process 504 enables the user to examine components, [such as variables including their type, class, and value], of the application program 304 to determine the source of problems with the application program 304." *Id.*, page 10, lines 9-13. "Figs. 4b-d show the debugger frame 410 with the various components in the application program 304, such as variables, variable values, fields, method calls, and constructors, used to generate the catalog web page I user frame 408" when the debug view option is selected. *Id.*, lines 7-9.

**Claim 17.**

Claims 17 is a means plus function type claim. The concise explanation of independent claim 17 follows directly from the explanation of claim 1 in that claim 17 claims means that perform the functions of the method of claim 1. Thus, the functions of each means plus function element of claim 17 are illustratively set forth in the description of claim 1. Every means plus

function element is identified and the structure, material, or acts described in the specification corresponding to each claimed means plus function element is set forth with reference to the specification in Table 1 below. All of the means plus function elements can be "embodied in the form of computer -implemented processes and apparatuses for practicing those processes." Present Application, page 3, lines 18-19.

| **Means Plus Function Element** | **Element** | **Specification and Figure References** |
|---|---|---|
| means for executing a web browser at the workstation | Client Workstation 112 and web browser 314. | Page 6, lines 2-3, 13-16;<br>Page 7, lines 27-29;<br>Figures 1 and 3. |
| means for invoking the application program and the debugger program using a user interface provided by the web browser and to cause the server to execute the application program and the debugger program | Client Workstation 112 and User Interface therein. | Page 6, lines 2-3, 13-16;<br>Page 8, lines 5-11;<br>Figures 1 and 3. |
| means for receiving a first web page from the server for displaying a user frame in the web browser at the workstation | Client Workstation 112 and web browser 314. | Page 8, lines 5-26;<br>Figures 1 and 3. |
| means for presenting the user frame of the first web page in the web browser at the workstation, wherein the user frame that includes information generated by the application program | Client Workstation 112 and web browser 314;<br>Graphical User Interface display 400;<br>User frame 408. | Page 8, line 21 - page 9, line 10;<br>Figures 1, 3, and 4a. |
| means for presenting a debug view option in the web browser at the workstation to generate a second web page having a debug frame of | Client Workstation 112 and web browser 314;<br>Graphical User Interface display 400; | Page 8, line 21 - page 9, line 10;<br>Figures 1, 3, and 4b-4d. |

| Means Plus Function Element | Element | Specification and Figure References |
|---|---|---|
| the application program | Debug frame 410. | |
| means for receiving the second web page from the server for displaying the debug frame in the web browser at the workstation when the debug view option is selected | Client Workstation 112 and web browser 314;<br>Graphical User Interface display 400;<br>Debug frame 410. | Page 8, line 27-page 9, line 22;<br>Figures 1, 3, and 4b-4d. |
| means for presenting the debug frame of the second web page in the web browser at the workstation when the debug view option is selected, wherein the debug frame includes information about one or more components of the application program | Client Workstation 112 and web browser 314;<br>Graphical User Interface display 400;<br>Debug frame 410. | Page 8, line 27-page 9, line 22;<br>Figures 1, 3, and 4b-4d. |

TABLE 1

**Claim 21.**

Claims 21 is a means plus function type claim. The concise explanation of independent claim 21 follows directly from the explanation of claim 9 in that claim 21 claims means that perform the functions of the method of claim 9. Thus, the functions of each means plus function element of claim 21 are illustratively set forth in the description of claim 9. Every means plus function element is identified and the structure, material, or acts described in the specification corresponding to each claimed means plus function element is set forth with reference to the specification in Table 2 below. All of the means plus function elements can be "embodied in the form of computer -implemented processes and apparatuses for practicing those processes." Present Application, page 3, lines 18-19.

| Means Plus Function Element | Element | Specification and Figure References |
|---|---|---|
| means for executing the application program and the debugger program on the server when the application program is invoked from the workstation | Server 114, 116, and/or 118;<br>Server 302. | Page 6, line 4;<br>Page 7, line 25 - page 8, line 11;<br>Figures 1 and 3. |
| means for generating information for a first web page, wherein the first web page comprises a user frame that includes information generated by the application program | Server 114, 116, and/or 118;<br>Server 302;<br>Application program 304. | Page 6, line 4;<br>Page 7, line 25 - page 8, line 4;<br>Page 8, lines 14-18;<br>Figures 1 and 3; |
| means for generating information for a second web page, wherein the second web page comprises a debug frame when a debug view option is selected from the workstation, wherein the debug frame includes information about components of the application program | Server 114, 116, and/or 118;<br>Server 302;<br>Debugger 312;<br>Processes 502-510. | Page 6, line 4;<br>Page 9, lines 14-22;<br>Page 9, line 26-page 11, line 11;<br>Figures 1, 3, and 5. |

TABLE 2

**Claim 29.**

With reference to the example embodiments of Figures 3, 4a, 4b, 4c, and 4d of the present application, the invention of claim 29 provides a "system for debugging an application program forma workstation, wherein the application program and a debugger program reside on a server that is remote from the workstation." Claim 29.

Referring to Figure 3, a web browser 314 is executed on a client workstation 112. An application program 304 executes on a server 302. Present Application, p. 7, lines 27-29. A

debugger program 312 also executes on server 302. *Id.*, p. 8, lines 5-11. The client workstation 112 and server 302 are remote from each other and connected to each other via a network interface 122. *Id.*, p. 6, lines 8-9. "Web browser 314 provides a user interface to access application programs 304 and debugger program 312 via network interface 122." *Id.*, p. 8, lines 7-8. "After [a] user enters [a uniform resource locator] into the workstation 112, the web browser 314 contacts the server via network interface 122, the corresponding application program 304 generates the requested web page, and the requested web page is transmitted via the network interface 122 to the web browser 314." *Id.*, lines 14-18. "Web browser 314 presents the requested web page to the user." *Id.*, line 18.

Referring to Figures 4a-d, the "GUI display 400 includes selectable features, such as tabs 404 and 406 to provide options to display a user frame 408 or a debugger frame 410." *Id.*, p. 9, lines 1-2. The "user frame 408 includes information generated by the application program 304 being debugged." *Id.*, lines 3-4. "For example, Fig. 4a shows a catalog web page for a color monitor in user frame 408." *Id.*, lines 6-7.

"For one of many reasons, user frame 408 may not render correctly." *Id.*, line 11. "To investigate the reasons for the problem, the developer selects the debugger view tab 406, thereby causing the web browser 314 to link to the debugger program 312." *Id.*, lines 11-13. "The debugger program 312 generates the debug information displayed in the debugger frame 410." *Id.*, lines 14-15. "Figs. 4b-d show the debugger frame 410 with the various components in the application program 304, such as variables, variable values, fields, method calls, and constructors, used to generate the catalog web page I user frame 408" when the debug view option is selected. *Id.*, lines 7-9. "A user can investigate the source of errors in the application program 304 that generates the information or error messages on the user frame 408." *Id.*, lines 15-16. "For example, user frame 408 can display the generic message "Null Pointer Exception" generated by the application program 304 when one or more variable(s) used by the web page has a null value." *Id.*, lines 16-19. "The debugger frame 410 allows the user to display the types and values of variables, methods and constructors associated with the variables, as shown in Figs 4b-4d." *Id.*, lines 19-21.

**Claim 36.**

Claim 36 includes both means plus function and non-means plus function elements. The concise explanation of the means plus function elements of claim 36 follows directly from the explanation of the first two elements of claim 9 in that the means plus function elements of claim 36 perform the functions of the first two elements of claim 9. Thus, the functions of each means plus function element of claim 21 are illustratively set forth in the description of claim 9. Every means plus function element is identified and the structure, material, or acts described in the specification corresponding to each claimed means plus function element is set forth with reference to the specification in Table 3 below. All of the means plus function elements can be "embodied in the form of computer -implemented processes and apparatuses for practicing those processes." Present Application, page 3, lines 18-19.

| **Means Plus Function Element** | **Element** | **Specification and Figure References** |
|---|---|---|
| means for executing the application program and the debugger program on the server when the application program is invoked from the workstation | Server 114, 116, and/or 118;<br>Server 302. | Page 6, line 4;<br>Page 7, line 25 - page 8, line 11;<br>Figures 1 and 3. |
| means for generating information for a first web page, wherein the first web page comprises a user frame that includes information generated by the application program | Server 114, 116, and/or 118;<br>Server 302;<br>Application program 304. | Page 6, line 4;<br>Page 7, line 25 - page 8, line 4;<br>Page 8, lines 14-18;<br>Figures 1 and 3; |

Claim 36 also includes a "debugger program operable to generate information for a second web page." Claim 36. "Server 302 also includes a debugger program 312 that can be used to troubleshoot problems from one or more remote client workstations 112." *Id.*, p. 8, lines 5-6. The debugger program 312 also executes on server 302. *Id.*, p. 8, lines 5-11. A "user frame

includes information generated by the application program 304 being debugged." *Id.*, 9, lines 3-4.

A debugger program "allows access to information pertaining to the application program being debugged." Page 5, lines 17-18. An example of when debugger 312 can be used is when a user, running web browser 314 on the workstation 112, enters the universal resource locator (URL) to access one of the application programs 304 for a web page." *Id.*, p. 8, lines 12-14. "After the user enters the URL into the workstation 12, the web browser 314 contacts the server 302 via the network interface 122, the corresponding application program 304 generates the requested web page, and the requested web page is transmitted via the network interface 122 to the web browser 314." *Id.*, lines 14-18. "The web page can include information from internal code 305, private code 308, and/or public code 310." *Id.*, lines 18-20.

"The debugger program 312 generates the debug information displayed in the debugger frame 410" that is transmitted to the workstation. *Id.*, p. 9, lines 14-15. Referring to Figures 3, 4b-4d, and 5, "Fig. 5 shows an embodiment of a process for generating the debugger frame 410 (Figs. 4b-4d) for an application program 304 (Fig. 3)." *Id.*, p. 9, lines 26-27. "The debugger frame 410 is mapped to the user frame 408 from the standpoint that the context of the debugger frame 410 is sensitive to the context of the user frame 408." *Id.*, lines 4-6. "In process 502, the current state of the application program 304 is stored." *Id.*, line 28. "Process 504 enables the user to examine components, [such as variables including their type, class, and value], of the application program 304 to determine the source of problems with the application program 304." *Id.*, page 10, lines 9-13. "Figs. 4b-d show the debugger frame 410 with the various components in the application program 304, such as variables, variable values, fields, method calls, and constructors, used to generate the catalog web page I user frame 408" when the debug view option is selected. *Id.*, lines 7-9.

**GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL - 37 CFR § 41.37(c)(1)(vi)**

**Claim Rejections - 35 U.S.C. § 103**

Claims 1-44 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,026,362 issued to Kim et al., (referred to herein as "*Kim*") in view of U.S. Patent

No. 6,938,245 issued to Spertus et al. (referred to herein as "*Spertus*"), and further in view of U.S. Patent No. 6,167,535 issued to Foote et al. (referred to herein as "*Foote*").

**ARGUMENT - 37 CFR § 41.37(c)(1)(vii)**

**Claim Rejections - 35 U.S.C. § 103**

Claims 1-44 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,026,362 issued to Kim et al., (referred to herein as "*Kim*") in view of U.S. Patent No. 6,938,245 issued to Spertus et al. (referred to herein as "*Spertus*"), and further in view of U.S. Patent No. 6,167,535 issued to Foote et al. (referred to herein as "*Foote*").

The Present Application identified a "requirement for the capability to access and debug application programs from a location that is remote from the site of the server that executes the application program(s)." Present Application, p. 1, lines 27-29.

Claim 1 relates to "debugging an application program from a workstation" using a web browser executing on a work station where "the application program reside[s] on a server that is remote from the workstation." Claim 1.

*Kim* relates to "an improved tool and method for debugging complex computer applications, [sic] displays the relationship between processes and resources of the processes and the contents of the stack and registers for threads of processes." *Kim*, Abstract.

*Spertus* relates to "An interactive system for debugging programs in which a persistent data base system responds to update queries containing debugging information from a debugging information source and to read queries on the debugging information from an interactive interface." *Spertus*, Abstract.

*Foote* relates to "[t]echniques for analyzing object-oriented computer programs." *Foote*, Abstract.

The Examiner admits that *Kim* "fails to explicitly state using a user interface provided by the web browser to invoke the application program and the debugger program." Office Action, 9/6/2006, page 3. The Examiner maintains that *Spertus* provides the missing teachings in col. 2,

lines 46-58. *Id.*, page 4. However, Applicants respectfully submit that *Spertus* teaches that "an executing program performs update queries to a debug data base system containing debugging information the source receives as a result of the execution of the program." *Spertus*, col. 2, lines 24-28. Applicants respectfully submit that *Kim* in view of *Spertus* and *Foote* fail to teach or suggest "invoking the application program and the debugger program using a user interface provided by the web browser and via a network interface to cause the server to execute the application program and the debugger program." Claim 1.

*Spertus* teaches that responding to a message "also involves performing a read query on the debug database and returning an HTML page that contains the results of the query." Figure 4 and col. 10, lines 16-31 of *Spertus* indicates that programs have already been executed and a web server client is used to select a particular "execution listing 413" which "also shows the time the debugging information for the program was recorded at 417." Thus, the web browser does not "invoke[e] the application program and the debugger program using a user interface provided by the web browser and via a network interface to cause the server to execute the application program and the debugger program" as required by Claim 1. The web browser of *Spertus* only allows retrieval of debugging information already executing or executed programs.

Applicants respectfully submit that this difference alone is sufficient for finding that Kim in view of *Spertus* and *Foote* fail to teach Claim 1 and, thus, sufficient for granting Applicants' appeal of Claim 1.

For at least similar reasons, Applicants respectfully

Similarly, Applicants respectfully submit that *Kim* in view of *Spertus* and *Foote* neither teaches nor suggests independent claims 9, 17, 21, 29, and 36 which recite in part:

> "executing the application program and the debugger program on the server when the application program is invoked from the workstation" (Claim 9);
>
> "means for invoking the application program and the debugger program using a user interface provided by the web browser and to cause the server to execute the application program and the debugger program" (Claim 17);

"means for executing the application program and the debugger program on the server when the application program is invoked from the workstation" (Claim 21);

"interact with a web page displayed by the web browser to allow a user to invoke the application program and the debugger program from the workstation to cause the server to execute the application program and the debugger program" (Claim 29); and

"means for executing the application program and the debugger program on the server when the application program is invoked from the workstation" (Claim 36).

Applicants respectfully submit that *Kim* in view of *Spertus* and *Foote* also fail to teach or suggest other aspects of the present invention.

The Examiner further admits that *Kim* and *Spertus* "fail to explicitly state receiving a first web page from the server for displaying a user frame in the web browser at the workstation; displaying the user frame of the first web page in the web browser at the workstation, wherein the user frame includes information generated by the application program." *Id.*, p. 4. Applicants respectfully submit that such lack of teaching by *Kim* and *Spertus* follows (and strengthens Applicants' position) from Applicants' submission that *Kim* in view of *Spertus* and *Foote* fail to teach or suggest "invoking the application program and the debugger program using a user interface provided by the web browser and via a network interface to cause the server to execute the application program and the debugger program" as required by Claim 1.

With regard to the lack of teaching or suggestion by *Kim* and *Spertus* regarding the particular "user frame … wherein the user frame includes information generated by the application program" of Claim 1, the Examiner further states that *Foote* "discloses this limitation in column 7, lines 45-56." Office Action, 9/6/2006, page 4. *Foote* teaches presenting a "hypertext document to a user" that contains responses to a query. *Foote*, col. 7, lines 38-41. The number of possible responses to the query are somewhat numerous. However, as discussed in more detail below, Applicants respectfully submit that the 'responses' relate to **information**

**regarding** active objects obtained during run-time of a Java program in contrast to "**information generated** by the application program" as required by claim 1.

*Foote* relates to "techniques for analyzing object-oriented computer programs" by storing and presenting to a user "information regarding active objects." *Foote*, Abstract and col. 6, lines 6-8.

*Foote* teaches that, "In order to analyze the execution of a Java program, the user executes the program with a Java virtual machine." *Foote*, col. 6, lines 2-3. "The virtual machine is responsible for interpreting the Java program, and compilation may be performed for increased efficiency." *Id.*, lines 3-6. Applicants respectfully submit that *Foote*, and thus, *Kim* in view of *Spertus* and *Foote*, fails to teach or suggest "displaying the user frame of the first web page in the web browser at the workstation, **wherein the user frame includes information generated by the application program**" as required by claim 1.

"FIG. 6 shows a process of storing information regarding active objects obtained during run-time of a Java program." *Id.*, lines 6-8. Applicants respectfully submit that the "information regarding active objects" is not "**information generated by the application program**" as required by claim 1.

*Foote* includes the following Figure descriptions. The Figures are of interest because they reflect a scope of "information regarding active objects", which Applicants respectfully submit demonstrates that the scope of teaching does not include "**information generated by the application program**" as required by claim 1:

> FIG. 6 shows a process of storing information regarding active objects obtained during run-time.
>
> FIG. 7 shows a process of presenting information regarding active objects obtained during run-time.
>
> FIGS. 8A-8E show a hypertext document of all classes that had instances that were active when the snapshot was taken.
>
> FIG. 9 shows a hypertext document of a class that had instances that were active when the snapshot was taken.
>
> FIG. 10 shows a hypertext document of instances of a class excluding subclasses.

FIG. 11 shows a hypertext document of instances of a class including subclasses.

FIG. 12 shows a hypertext document of information regarding an active object.

FIG. 13 shows a hypertext document of root set references to an active object that excludes weak references.

FIG. 14 shows a hypertext document of root set references to an active object that includes weak references.

FIGS. 15A and 15B show a hypertext document of all objects reachable from an active object.

FIGS. 16A-16E show a hypertext document of all members of the root set.

Applicants respectfully submit that none of the information **regarding** active objects is "information **generated** by the application program" as required by claim 1.

A review of the Figures clearly supports Applicants' position.

"FIG. 6 shows a process of storing information regarding active objects obtained during run-time of a Java program." *Foote*, col. 6, lines 6-8. The "stored information regarding active objects … may include not only information about the active objects themselves, but also their relationships to each other." "The relationships between active objects may resemble a directed graph structure that begins with members of the root set and typically proceeds to other objects through various levels of indirection. *Id.* lines 54-60. Thus, the information regarding active objects is not "**information generated by the application program**" as required by claim 1. "FIG. 7 shows a process of presenting information regarding active objects obtained during run-time." *Id.*, lines 52-53. "At a step 451, the system retrieves stored information regarding active objects." *Id.*, lines 52-55. "The information may include not only information about the active objects themselves, but also their relationships to each other." *Id.*, lines 55-57. Again, the information regarding active objects is not "**information generated by the application program**" as required by claim 1.

"Once the information regarding the active objects is retrieved, the system receives a user query at a step 453." *Id.*, lines 64-65. "In a preferred embodiment, the query is specified as a

Uniform Resource Locator (URL) into a HyperText Transport Protocol (HTTP) server." *Id.*, col. 6, line 65-col. 7, line 1.

The following describe queries that may be available and the information regarding active objects that such queries obtain. Notably, none of the information regarding the active objects is "**information generated by the application program**" as required by claim 1:

> An "All Classes Query" shows all of the classes that were present on the heap at run-time. The classes may be sorted by their fully-qualified class name and organized by package. An example of the results of this query is shown in FIGS. 8A-8E.
>
> A "Class Query" shows information about a desired class. The information may include the superclass, any subclasses, instance data members, and static data members. An example of the results of this query are shown in FIG. 9.
>
> An "Instances Query" shows all the instances of a specified class. An example of the results of this query are shown in FIG. 10.
>
> An "Object Query" shows information about an object that was on the heap at run-time. Most notably, one may navigate to objects that refer to this object, which may be utilized to track down errors. An example of the results of this query are shown in FIG. 12.
>
> A "Roots Query" provides the reference chains from to root set to a specific object. A chain will be provided from each member of the root set from which the object of interest is reachable. In preferred embodiments, the chains are calculated by a depth-first search in order to reduce the length of the chains. Other search techniques may also be utilized. The "Roots Query" is very valuable query for tracking down memory leaks as it may be utilized to determine why an object is still active. An example of the results of this query are shown in FIG. 13.
>
> A "Reachable Objects Query" shows the transitive closure of all objects that are reachable from a specific object. This query may be useful for determining the total run-time footprint of an object in memory. An example of the results of this query are shown in FIG. (after FIG. 12).
>
> An "All Roots Query" shows all the members of the root set. An example of the results of this query are shown in FIGS. 16A-16E.
>
> *Foote*, col. 7, lines 3-37.

Thus, the HTML document generated by *Foote* and referenced by the Examiner is not a "user frame of the first web page in the web browser at the workstation, wherein the user frame

includes information generated **by the application program**" as required by claim 1. Rather, the HTML document generated by *Foote* represents:

> (i) "classes that were present on the heap at run-time" as shown in Figures 8A-8E;
>
> (ii) "information about a desired class" as shown in Figure 9;
>
> (iii) "all the instances of a specified class" as shown in Figure 10;
>
> (iv) "information about an object that was on the heap at run-time" as shown in Figure 12;
>
> (v) "reference chains from to root set to a specific object" as shown in Figure 13; and
>
> (vi) "the transitive closure of all objects that are reachable from a specific object" as shown in "FIG. (after FIG. 12)", (vii) "all the members of the root set" as shown in Figures 16A-16E.
>
> *Foote*, col. 7, lines 8-37.

Accordingly, Applicants respectfully submit that *Kim* in view of *Spertus* and *Foote* neither teaches nor suggests claim 1.

Similarly, Applicants respectfully submit that *Kim* in view of *Spertus* and *Foote* neither teaches nor suggests independent claims 9, 17, 21, 29, and 36 which recite in part:

> … [a] first web page … , wherein the first web page comprises a user frame that includes information generated by the application program.

Accordingly, for at least the foregoing reasons Applicants respectfully submit that *Kim* in view of *Spertus* and *Foote* fail to teach or suggest the present invention of claims 1, 9, 17, 21, 29, and 36 and claims directly or indirectly dependent upon claims 1, 9, 17, 21, 29, and 36

**CLAIMS APPENDIX - 37 CFR § 41.37(c)(1)(viii)**

A copy of the pending claims involved in the appeal is attached as Claims Appendix.

**EVIDENCE APPENDIX - 37 CFR § 41.37(c)(1)(ix)**

There is no evidence relied upon in the appeal.

**RELATED PROCEEDINGS APPENDIX - 37 CFR § 41.37(c)(1)(x)**

There are no related proceedings.

## CONCLUSION

For at least the reasons set forth above, Applicants respectfully submit that claims 1-44 are allowable. Accordingly, Applicants respectfully request withdrawal of the rejections of claims 1-44.

ELECTRONICALLY FILED

Respectfully submitted,

*/Kent B. Chambers/*

Kent B. Chambers
Attorney for Applicant
Reg. No. 38,839

1

## CLAIMS APPENDIX

1. (Previously Presented) A method of debugging an application program from a workstation, wherein the application program and a debugger program reside on a server that is remote from the workstation, the method comprising:

executing a web browser at the workstation;

invoking the application program and the debugger program using a user interface provided by the web browser and via a network interface to cause the server to execute the application program and the debugger program;

receiving a first web page from the server for displaying a user frame in the web browser at the workstation;

displaying the user frame of the first web page in the web browser at the workstation, wherein the user frame includes information generated by the application program;

displaying a debug view option in the web browser at the workstation for generating a second web page having a debug frame of the application program;

receiving the second web page from the server for displaying the debug frame in the web browser at the workstation when the debug view option is selected; and

displaying the debug frame of the second web page in the web browser at the workstation, wherein the debug frame includes information about one or more components of the application program.

2. (Original) The method of Claim 1 further comprising:

providing a user view option at the workstation for generating the user frame; and

displaying the user frame when the user view option is selected.

3. (Previously Presented) The method of Claim 1 wherein displaying the debug frame of the second web page in the web browser at the workstation includes providing a list of variable names in the application program.

4. (Previously Presented) The method of Claim 3 wherein displaying the debug frame of the second web page in the web browser at the workstation includes providing at

least one of: a list of request information variable names in the application program, or a list of session information variable names in the application program.

5. (Original) The method of Claim 3, wherein one or more of the variable names represents a corresponding object, the method further comprising:

selecting one of the variable names; and

providing information about the object corresponding to the variable name on the debug frame.

6. (Original) The method of Claim 5 wherein the information about the object includes at least one of: the fields of the object, the methods associated with the object, or the constructors associated with the object.

7. (Original) A computer readable storage media comprising:

computer instructions to implement the method of claim 1.

8. (Original) A signal in a carrier medium comprising:

computer instructions to implement the method of claim 1.

9. (Previously Presented) A method of debugging an application program from a workstation, wherein the application program and a debugger program reside on a server that is remote from the workstation, the method comprising:

executing the application program and the debugger program on the server when the application program is invoked from the workstation;

generating information for a first web page, wherein the first web page comprises a user frame that includes information generated by the application program;

transmitting the first web page to the workstation;

generating information for a second web page, wherein the second web page comprises a debug frame when a debug view option is selected from the workstation, wherein the debug frame includes information about components of the application program; and

transmitting the second web page to the workstation.

10. (Previously Presented) The method of Claim 9 wherein generating information for second web page includes saving the information for the user frame when the debug view option is selected.

11. (Original) The method of Claim 10 further comprising restoring the saved information for the user frame when a user view option is selected at the workstation.

12. (Previously Presented) The method of Claim 9 wherein generating information for the second web page includes generating a list of components of the application program.

13. (Previously Presented) The method of Claim 9 wherein generating information for the second web page includes generating at least one of: a list of variables in the application program, a list of methods associated with one or more of the variables in the application program, or a list of constructors associated with one or more of the variables in the application program.

14. (Currently Amended) The method of Claim 9 wherein generating information for the second web page [[e]] includes using reflection technology to generate at least one of: a list of variables in the application program, a list of methods associated with one or more of the variables, and a list of constructors associated with one or more of the variables.

15. (Original) A computer readable storage media comprising:
computer instructions to implement the method of claim 9.

16. (Original) A signal in a carrier medium comprising:
computer instructions to implement the method of claim 9.

17. (Previously Presented) An apparatus for debugging an application program from a workstation, wherein the application program and a debugger program reside on a server that is remote from the workstation, the method comprising:
means for executing a web browser at the workstation;

means for invoking the application program and the debugger program using a user interface provided by the web browser and to cause the server to execute the application program and the debugger program;

means for receiving a first web page from the server for displaying a user frame in the web browser at the workstation;

means for presenting the user frame of the first web page in the web browser at the workstation, wherein the user frame that includes information generated by the application program;

means for presenting a debug view option in the web browser at the workstation to generate a second web page having a debug frame of the application program;

means for receiving the second web page from the server for displaying the debug frame in the web browser at the workstation when the debug view option is selected; and

means for presenting the debug frame of the second web page in the web browser at the workstation when the debug view option is selected, wherein the debug frame includes information about one or more components of the application program.

18. (Original) The apparatus of Claim 17 further comprising:

means for presenting a user view option at the workstation for generating the user frame; and

means for presenting the user frame when the user view option is selected.

19. (Previously Presented) The apparatus of Claim 17 wherein the debug frame of the second web page in the web browser at the workstation includes a list of components of the application program.

20. (Original) The apparatus of Claim 17, wherein one or more of the components represents a corresponding object, the apparatus further comprising:

means for selecting one of the objects; and

means for presenting information about the selected object, wherein the information about the object includes at least one of: the name of the object, the fields of the object, the methods associated with the object, or the constructors associated with the object.

21. (Previously Presented) An apparatus for debugging an application program from a workstation, wherein the application program and a debugger program reside on a server that is remote from the workstation, the method comprising:

means for executing the application program and the debugger program on the server when the application program is invoked from the workstation;

means for generating information for a first web page, wherein the first web page comprises a user frame that includes information generated by the application program; and

means for generating information for a second web page, wherein the second web page comprises a debug frame when a debug view option is selected from the workstation, wherein the debug frame includes information about components of the application program.

22. (Previously Presented) The apparatus of Claim 21 further comprising means for saving the information for the user frame when the debug view option is selected.

23. (Original) The apparatus of Claim 22 further comprising means for restoring the saved information for the user frame when a user view option is selected at the workstation.

24. (Previously Presented) The apparatus of Claim 21 further comprising means for generating a list of objects in the application program.

25. (Previously Presented) The apparatus of Claim 24 wherein the means for generating a list of objects in the application program includes at least one of: means for generating a list of methods associated with one or more of the objects in the application program, or means for generating a list of constructors associated with one or more of the objects in the application program.

26. (Previously Presented) The apparatus of Claim 24 further comprising means for using reflection technology to generate at least one of: a list of objects in the application program, a list of methods associated with one or more of the objects, and a list of constructors associated with one or more of the objects.

27. (Previously Presented) The apparatus of Claim 24 further comprising:

means for providing the list of objects to the workstation in a web page when the debug view option is selected at the workstation.

28. (Previously Presented) The apparatus of Claim 25 further comprising:

means for providing in a web page at least one of: a list of names of the objects, a list of fields of at least one of the objects, a list of values of at least one of the objects, the list of methods associated with at least one of the objects, or the list of constructors associated with at least one of the objects.

29. (Previously Presented) A system for debugging an application program from a workstation, wherein the application program and a debugger program reside on a server that is remote from the workstation, the system comprising:

a web browser executable at the workstation operable to:

interact with a web page displayed by the web browser to allow a user to invoke the application program and the debugger program from the workstation to cause the server to execute the application program and the debugger program;

present a first web page in the web browser, wherein the first web page comprises a user frame that includes information generated by the application program;

present a debug view option to generate a second web page having a debug frame of the application program; and

present the debug frame of the second web page when the debug view option is selected, wherein the debug frame includes information about one or more components of the application program.

30. (Previously Presented) The system of Claim 29 wherein the web browser executable at the workstation is further operable to:

present a user view option at the workstation; and

present the user frame when the user view option is selected.

31. (Original) The system of Claim 29 wherein the debug frame at the workstation includes a list of one of more components of the application program.

32. (Previously Presented) The system of Claim 29 wherein the application program generates instructions and information for displaying the web pages.

33. (Previously Presented) The system of Claim 29 wherein web browser is operable to display graphical user controls to allow the workstation to communicate with the server.

34. (Previously Presented) The system of Claim 31 further wherein the web browser is operable to present a third web page, and the third web page comprises additional information about at least one of the components when the component is selected by the user.

35. (Original) The system of Claim 34, wherein the additional information includes at least one of: the name of the component, the fields of the component, the methods associated with the component, or the constructors associated with the component.

36. (Previously Presented) A system for debugging an application program from a workstation, wherein the application program and a debugger program reside on a server that is remote from the workstation, the system comprising:

means for executing the application program and the debugger program on the server when the application program is invoked from the workstation;

means for generating information for a first web page, wherein the first web page comprises a user frame that includes information generated by the application program; and

a debugger program operable to generate information for a second web page, wherein the second web page comprises a debug frame when a debug view option is selected from the workstation, wherein the debug frame includes information about components of the application program.

37. (Original) The system of Claim 36 wherein the debugger program is operable to save the information for the user frame when the debug view option is selected.

38. (Original) The system of Claim 37 wherein the debugger program is operable to restore the saved information for the user frame when a user view option is selected at the workstation.

39. (Original) The system of Claim 36 wherein the debugger program is operable to generate a list of objects in the application program.

40. (Previously Presented) The system of Claim 39 wherein the debugger program is operable to generate in a web page at least one of: a list of methods associated with one or more of the objects in the application program, or a list of constructors associated with one or more of the objects in the application program.

41. (Original) The system of Claim 36 wherein the debugger program is operable to use reflection technology to generate at least one of: a list of objects in the application program, a list of methods associated with one or more of the objects, and a list of constructors associated with one or more of the objects.

42. (Previously Presented) The system of Claim 36 wherein the application program generates instructions and information for displaying the web pages.

43. (Original) The system of Claim 36 wherein the server is operable to communicate with a web browser program at the workstation.

44. (Original) The system of Claim 36 wherein the application program accesses at least one of: internal code, private code, or public code.

**EVIDENCE APPENDIX - 37 CFR § 41.37(c)(1)(ix)**

None

0

**RELATED PROCEEDINGS APPENDIX - 37 CFR § 41.37(c)(1)(x)**

None